Radio Frequency Fingerprints vs. Physical Unclonable Functions - Are They Twins, Competitors or Allies?

Junqing Zhang, Chip-Hong Chang, *Fellow, IEEE*, Chongyan Gu, *Senior Member, IEEE*, and Lajos Hanzo, *Fellow, IEEE*

***Abstract*—Privacy breaches and online frauds are grave concerns in pervasive computing. Device identification is the first line of defense to detect and stop fraud. Conventional device authentication schemes using software addresses as identities or static pre-programmed secret keys are vulnerable to tampering and memory attacks. This article reviews two emerging lightweight hardware-oriented solutions to avoid these problems, namely radio frequency fingerprint (RFF) identification and physical unclonable function (PUF) authentication. Their operating principles and protocols are first introduced, followed by a scrutiny of their common and distinctive features, and a discussion of the stumbling blocks in the way of their market adoption. Finally, we envisage a combined mutual authentication and key establishment scheme to shed light on their synergy.**

***Index Terms*—Internet of things, security, lightweight authentication, radio frequency fingerprint, physical unclonable function**

I. Introduction

The Internet of things (IoT) has transformed our daily life, supported by Billions of connected devices. It is predicted that by 2030, there will be 500 billion connected devices, ranging from consumer-oriented devices, enterprise equipment to industrial assets[1]. At this growth rate, it is challenging to safeguard their applications due to their ubiquity, heterogeneity, resource constraints and accessibility. To ensure that only trusted devices can gain access to a network, high-security device identification mechanisms have to replace the publicly known unique device IDs or node addresses.

Conventional device authentication schemes rely on computational cryptography for verifying the device identity, typically by solving a puzzle (cryptogram) involving a shared secret in a challenge-response protocol. Software addresses such as the Media Access Control (MAC) address are often used to identify the source of a packet in communication networks. However, a software address is never a reliable identifier, since it can be easily sniffed and tampered with by an attacker. Many incidents of leaking sensitive data through address resolution protocol spoofing/poisoning have been reported. The MAC credentials have to be secured for example by 802.1X or MACsec, both of which use a shared key established through public-key cryptography (PKC). Unfortunately, PKCs are too complex and hence power-hungry for IoT devices. In addition to secure key storage, some cryptographic solutions also need regular firmware updates, but this turns out to be challenging for many legacy devices.

The pitfalls of relying on heavy-weight PKC for shared key establishment or costly secure non-volatile key storage in classical cryptography motivates us to seek alternative covert device identities and their identification. The identity should ideally be generated upon request or when the device is powered up to perform the required operations. Due to the unavoidable semiconductor fabrication process variability, no pair of IoT devices are identical, even if they are produced using the same transistor model by the same manufacturer in the same production lot. These non-volatile features are omnipresent and intrinsic in nano-scale devices. Owing to their subtle structural irregularities, these tamper-reactive innate traits can be exploited to identify a device uniquely.

Two such popular "keyless" device identification techniques are constituted by radio frequency fingerprints (RFFs) and physical uncolonable functions (PUFs). Specifically, an RFF identifies a wireless device by its unique transmission characteristics due to slight deviations in nonlinear radio frequency (RF) component values, such as oscillator drift, in-phase (I) and quadrature-phase (Q) imbalances and power amplifier (PA) non-linearity [1], [2]. These hardware impairments fall within the tolerance of their component specifications. Hence they do not affect the normal communication functionality, but the transmitted waveforms still exhibit unique characteristics. By contrast, a PUF realizes a random oracle by harnessing the physical disorder arising from inter- and intra-die geometry mismatch in the semiconductor device fabrication process [3]. The lynchpin of PUF is the irreversible random mapping of a digital input to a digital output. It is in many respects reminiscent of a cryptographic hash function except that the

Manuscript received xxx; revised xxx; accepted xxx. Date of publication xxx; date of current version xxx. J. Zhang and C. H. Chang contributed equally to this work.

J. Zhang is with the Department of Electrical Engineering and Electronics, University of Liverpool, Liverpool, L69 3GJ, United Kingdom. (email: junqing.zhang@liverpool.ac.uk)

C. H. Chang is with the School of Electrical and Electronic Engineering, Nanyang Technological University (NTU), Singapore, 639798. (e-mail: echchang@ntu.edu.sg)

C. Gu is with the Centre for Secure Information Technologies (CSIT), Institute of Electronics, Communications & Information Technology (ECIT), Queen's University Belfast (QUB), U.K., BT3 9DT. (e-mail: c.gu@qub.ac.uk, cgu01@qub.ac.uk)

L. Hanzo is with the School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, U.K. (email: lh@ecs.soton.ac.uk)

Digital Object Identifier xxx

[1]https://www.cisco.com/c/en/us/solutions/collateral/data-center-virtualization/big-data/solution-overview-c22-740248.html

hardness to invert the function is originated from physical disorder instead of computational complexity theory. A PUF can generate not only a set of attributes for chip identification, but also serve as a hardware cryptogram to unleash new options for IoT security, as exemplified in [3]–[5].

An RFF extractor or PUF primitive also affiliates a device with a specific trusted group or network. Although both RFF and PUF exploit the random manufacturing process variations for enhancing security, there are subtle differences between them and both have their unique challenges as well as limitations. Over the last decade, substantial efforts have been made to promote both RFF [1] and PUF [3] in isolation, but no attempt has been made to critically appraise or beneficially combine them.

II. Conventional Cryptography-Based Authentication

Cryptography-based authentication protocols are widely adopted in commercial wireless technologies. The device authentication scheme of a popular Long Range Wide Area Network specification, LoRa/LoRaWAN® for low-power wireless connectivity is introduced as an example. LoRa defines the physical layer modulation, while LoRaWAN specifies the network architecture and MAC layer protocol.

LoRaWAN v1.1 specifies two device admission procedures, namely over the air activation (OTAA) and activation by personalization (ABP). OTAA is more secure than ABP. In OTAA, the LoRa end devices and servers have two 128-bit pre-shared root keys, namely NwkKey and AppKey. Each LoRa device is assigned a globally unique 64-bit Device Extended Unique Identifier (DevEUI) and a JoinEUI that is a 64-bit global application ID for identifying the device and its server, respectively. To join a network, the device transmits an unencrypted join-request message, which consists of JoinEUI, DevEUI and a 16-bit DevNonce. DevNonce is a counter value, which is incremented with every join-request to prevent replay attacks. A message integrity code (MIC) is calculated for the join-request message by the Advanced Encryption Standard (AES) using the NwkKey. Upon receiving the join-request, the network server consults the application server associated with JoinEUI to validate the request. If permission is granted, it responds with a join-accept message comprising JoinNonce and other information. The JoinNonce will be used to generate new session keys for the encryption and decryption of the payload between the device and the application server or for validating the MICs of session messages.

Similar to other software address-based schemes, OTAA is vulnerable to spoofing attacks. Because Join-request is not encrypted, any attackers can decode the message to recover the pre-programmed DevEUI in the device memory. Once the NwkKey is stolen, a spoofing attack can be initiated. Unfortunately, the reliance on secret keys is the Achilles heel of cryptographic authentication schemes. If the keys are stored off-chip, they must be fetched over the memory bus when needed. Protection of plaintext transmission via the memory bus is expensive and has limited efficiency. Alternatively, the keys can be kept on chip in non-volatile memory (NVM). The problem with NVM is that its content can be accessed by physical attacks when the power is disconnected. If a battery-backed static random access memory (SRAM) is used instead, the keys can be erased when the on-chip tamper-detection sensors detect a physical attack. The memory content will be lost if the battery is removed to disable the sensors. Besides the cost and size of the permanent battery, a major drawback of using volatile memory for key preservation is the memory "imprint" can be retrieved from powered-down cells by data remanence or cold boot attacks.

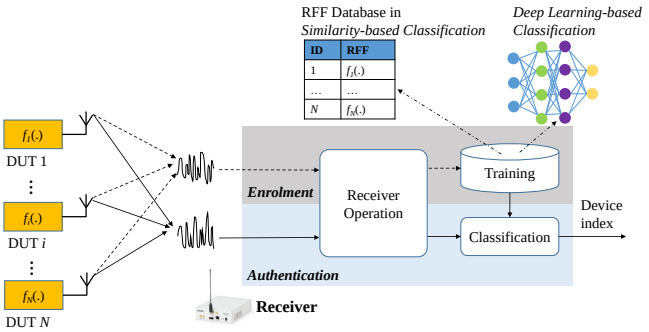

Fig. 1. RFF identification protocol.

III. Radio Frequency Fingerprint Identification

A. Concept

RFF exploits hardware imperfections of transmitters to authenticate wireless devices. Since the radio signals are uniquely shaped by the nonlinear analog components of their transmitters, it is possible to extract unique features to distinguish different transmitters from their RF waveforms.

Fig. 1 shows N wireless devices under test (DUTs) and a receiver that acts as an authenticator. The signal received from the i^{th} DUT can be written as

$$y_i(t) = h(t) * f_i(x) + n(t), \tag{1}$$

where $h(t)$ is the wireless channel, $*$ denotes the convolution operation, $f_i(\cdot)$ is the overall effect of hardware distortion on the transmitted analog signal x, and $n(t)$ is the noise.

The receiver learns and stores the hardware features of the N DUTs from their physical layer waveforms during the enrollment phase. In the authentication phase, the receiver infers and analyzes the transmitter features from the received waveforms. The transmitter can then be identified by matching the extracted features against the feature library.

B. RFF Features

The intrinsic imperfections of an RF transmitter include I/Q imbalances, frequency drift of oscillators and PA non-linearity, etc. [2], as illustrated in Fig. 2. The transmitted signal flowing through the hardware chain will be distorted by the parametric deviations of these components, which form the material basis of $f_i(\cdot)$ in (1).

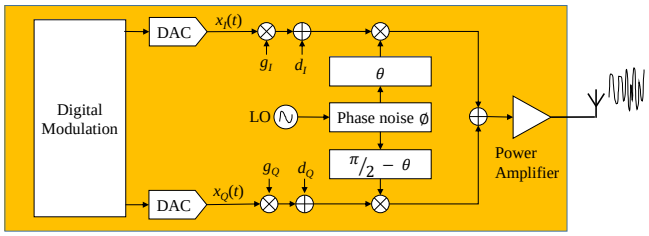

Fig. 2. Hardware architecture of wireless transceiver.

1) IQ Gain and Phase Imbalance: The transceiver uses the quadrature mixer to upconvert baseband signals or downconvert RF signals. The signals are processed in the complex domain. The I and Q branches may have different gains, g_I and g_Q, DC offsets, d_I and d_Q, as well as phase mismatch θ.

2) Oscillator Imperfection: The frequency generated by the local oscillator (LO) typically deviates from the nominal carrier frequency, f_c, by a carrier frequency offset (CFO) of Δf. The LO is also subject to phase noise.

3) PA Non-linearity: The PA boosts the power of the RF signal to the desired output level. In narrowband systems, the PA experiences memoryless effects, which are often modeled as amplitude/amplitude (AM/AM) and amplitude/phase (AM/PM) characteristics. In wideband systems, the PA experiences memory-effect distortions, whereby current symbols are distorted by prior symbols, usually characterized by polynomial models.

C. Protocol

The RFF protocol is portrayed in Fig. 1. In the enrollment phase, the DUTs emit signals in turn; the receiver collects these physical waveforms to train a classifier. After enrollment, the receiver can promptly predict the class label of a DUT from its waveform to achieve per-packet authentication. There are two main categories of classification schemes.

1) Similarity-based Classification: During enrollment, the receiver uses existing parts, e.g., preambles, in the received packets to estimate the unique features, such as CFO and IQ mismatch, of the DUTs. The features are saved as templates in an RFF database. During authentication, the same features are extracted by the receiver from the received packet for comparison against the templates in the database to infer the identity of the DUT based on their similarity.

A standard learning problem in Euclidean space can be formulated by treating the similarities between an input sample and the training samples. Its generalization performance can be improved by a hybrid classifier [6], which weighs each feature differently according to the signal-to-noise ratio (SNR) during training. During matching, the selected features are combined adaptively according to the estimated SNR.

2) Artificial Intelligence-based Classification: RFF identification (RFFI) is a multi-class classification problem. It is therefore not surprising that machine learning (ML) and deep learning (DL) solutions have gained popularity in recent years.

In an ML-based scheme, handcrafted features are extracted from the received signals to train an ML model such as a support vector machine (SVM) or random forest. The seminal work in [7] extracted RFF features such as CFO, IQ offset, phase error and used k-nearest-neighbor and SVM classifiers. It achieved 99% accuracy in classifying 138 WiFi devices.

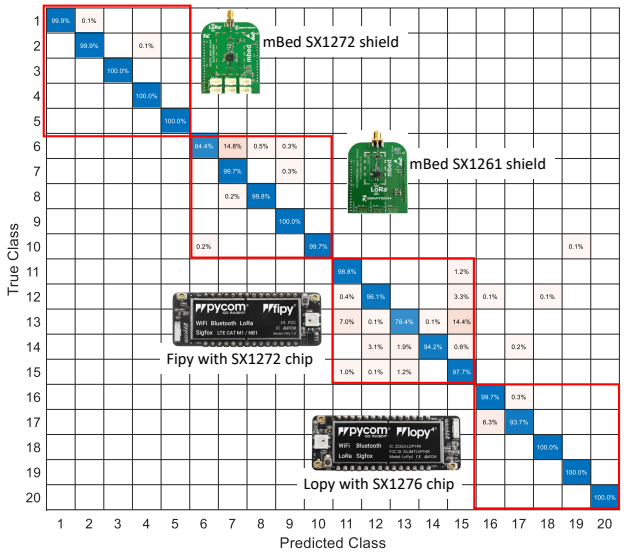

Fig. 3. Confusion matrix of the LoRa-based RFFI. 20 LoRa devices of four models were used. There were 1000 test packets for each device.

In a recently reported DL-based scheme [8], the receiver directly uses the IQ samples, $y_i(t)$, to train a DL network, as illustrated in Fig. 1. The RFF features are intrinsically embedded in the IQ samples. The identification performance can be enhanced by signal preprocessing.

D. Case Study of LoRa RFFI

A deep learning-based LoRa-RFFI system is exemplified [9]. We used 20 LoRa DUTs of four models and a Universal Software Radio Peripheral (USRP) N210 as the receiver. IQ samples of LoRa signals are transformed to spectrograms for visualizing their time-frequency characteristics. As the spectrogram is an image, it can be processed by a convolutional neural network (CNN) efficiently. An overall accuracy of 97.10% was achieved by using a low-complexity CNN architecture having three convolutional layers. The resultant classification matrix is shown in Fig. 3. It is observed that the misclassifications mainly occur within the 5×5 squares. The DUTs in each square are of the same model. Hence, their RFF features have higher similarities and are harder to discriminate.

IV. Physical Unclonable Function

A. Concept

A PUF is a physical system that reacts to an input stimulus (challenge c) in a complex and unpredictable way to produce a measurable output (response r), which can be expressed as:

$$r = PUF(c). \tag{2}$$

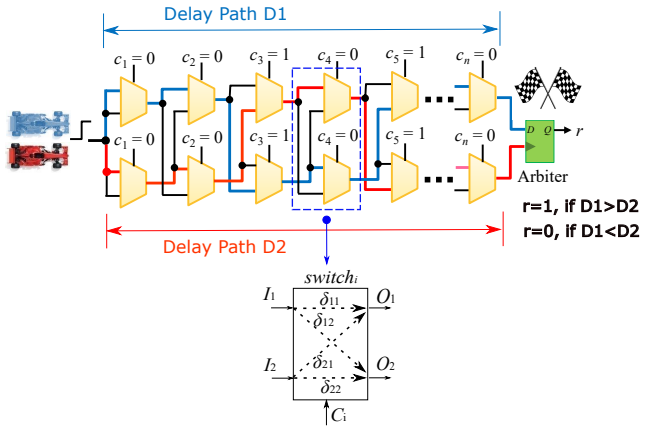

Fig. 4. An arbiter PUF with n challenge bit cells [10].

This challenge-response mapping can be gleaned from an integrated circuit that contains nanoscale elements with an abundance of delicate structural information. It is impossible to control the geometric parameters of these elements precisely at manufacturing time that make each PUF instance unique and physically unclonable. A PUF cell is designed for harnessing the differences in the electrical properties - such as resistance, current, voltage, propagation delay, etc. - arising from the minute geometric mismatches between symmetric circuits from chip to chip, or even from adjacent transistors within the same chip. The PUF cells are activated by a digital input (challenge) to extract the electrical differences. The resultant analog signals are then aggregated or amplified before they are digitized into a multi-bit response by a differential amplifier, counter, comparator, arbiter, etc. Given a sufficient number of basic PUF cells, the chip-to-chip variances of a manufacturing process can be exploited for producing a unique response of arbitrary length from each chip for a huge number of manufactured chips. Although the PUF circuit itself is easy to make and the responses can be readily measured, it is practically impossible to physically clone a PUF instance for reproducing the same challenge-response behavior.

B. PUF Architecture

Diverse architectures can be used for constructing a multi-bit response PUF. These architectures can be broadly categorized into strong and weak PUFs according to their number of unique challenges. It is important to ensure for device authentication that the responses to all possible challenges of a PUF cannot be exhaustively measured within a limited time. To keep the PUF lightweight, strong PUF is preferred as its number of unique challenges increases exponentially with the number of basic cells.

The Arbiter PUF [10] is the most classic strong PUF, which is shown in Fig. 4. It has n cascaded stages of challenge bit cells. Each stage consists of a pair of multiplexers configured as a cross-bar switch. The select inputs of the two multiplexers at Stage i are both fed by C_i, where C_i is the i-th bit of an n-bit challenge. The switch at Stage i has a parallel connection if $C_i = 0$, and a criss-cross connection otherwise. Altogether there are 2^n different symmetrical paths selectable by an n-bit challenge. The bottom subfigure in Fig. 4 shows the delays of the two paths through the switch at Stage i, i.e., I_1 to O_1 (δ_{11}) and I_2 to O_2 (δ_{22}) when C_i is 0, and I_1 to O_2 (δ_{12}) and I_2 to O_1 (δ_{21}) when C_i = 1. The output signals of the two multiplexers at the last stage are connected respectively to the data and clock inputs of a D Flip-flop, which acts as an arbiter to determine which of the signals along the two delay paths D1 (blue) and D2 (red) is faster. The arbiter outputs a logical 1 when the delay path D2 is faster than the delay path D1 and 0 otherwise. The delays δ_{11}, δ_{12}, δ_{21} and δ_{22} of each switch between two different PUF instances are slightly different due to the manufacturing process variation. Hence, a different output may be obtained by applying the same challenge C to two different PUF instances. Each application of an n-bit challenge produces only a single response bit. To generate an l-bit response, the n-bit challenge is fed into an n-bit linear feedback shift register (LFSR). Then, the PUF is evaluated l times using l different pseudo-random outputs of the LFSR as challenges.

C. Protocol

A lightweight PUF-based mutual authentication protocol [11] is illustrated in Fig. 5. During enrollment, the server sends a challenge c_i to the DUT, and the DUT replies by returning a multi-bit response r_i from its embedded PUF. A sufficiently large number of challenge-response pairs (CRPs) are then collected for future authentication and stored in the server's database *DB* alongside the public device *ID*, as seen in Fig. 5. For security, the interface for direct CRP measurement will be disabled after enrollment. The DUT is equipped with a reverse fuzzy extractor [11] and a lightweight hash function. A fuzzy extractor consists of a secure sketch and a randomness extractor for enhancing the reliability and randomness of the PUF response, respectively. The secure sketch generates the helper data $h_i = \mathit{Gen}(r_i')$ from the noisy PUF response r_i' for recovering the clean response $r_i = \mathit{Rep}(r_i', h_i)$. The reverse fuzzy extractor in Fig. 5 offloads the complex error decoding function $\mathit{Rep}()$ to the server by reproducing the noisy r_i' instead from the clean r_i with the aid of h_i.

Still referring to Fig. 5, when the server initiates an authentication request to a DUT, the DUT responds with its public *ID*. If *ID* exists in *DB*, the server chooses a nonce N and a random CRP (c_i, r_i) of *ID* from *DB*. Only c_i and N are sent to the DUT. The DUT feeds c_i into its PUF to produce r_i'. It then computes $h_i = \mathit{Gen}(r_i')$ and $a = hash(\mathit{ID}, N, r_i', h_i)$ and sends them to the server. The server reproduces $\hat{r} = \mathit{Rep}(r_i, h_i)$ and computes $hash(\mathit{ID}, N, \hat{r}, h_i)$ to check against a. If they do not match, the session is aborted. Otherwise, the server accepts the DUT by sending it $b = hash(a, \hat{r})$. The DUT computes $hash(a, r_i')$ to check against b. If they match, it accepts the server. Otherwise, the session is aborted.

Table I lists the figures of merits of four different 64-bit strong PUFs implemented on either field-programmable gate array (FPGA) or application-specific integrated circuit (ASIC) platforms. Any of these PUFs can be used in the above

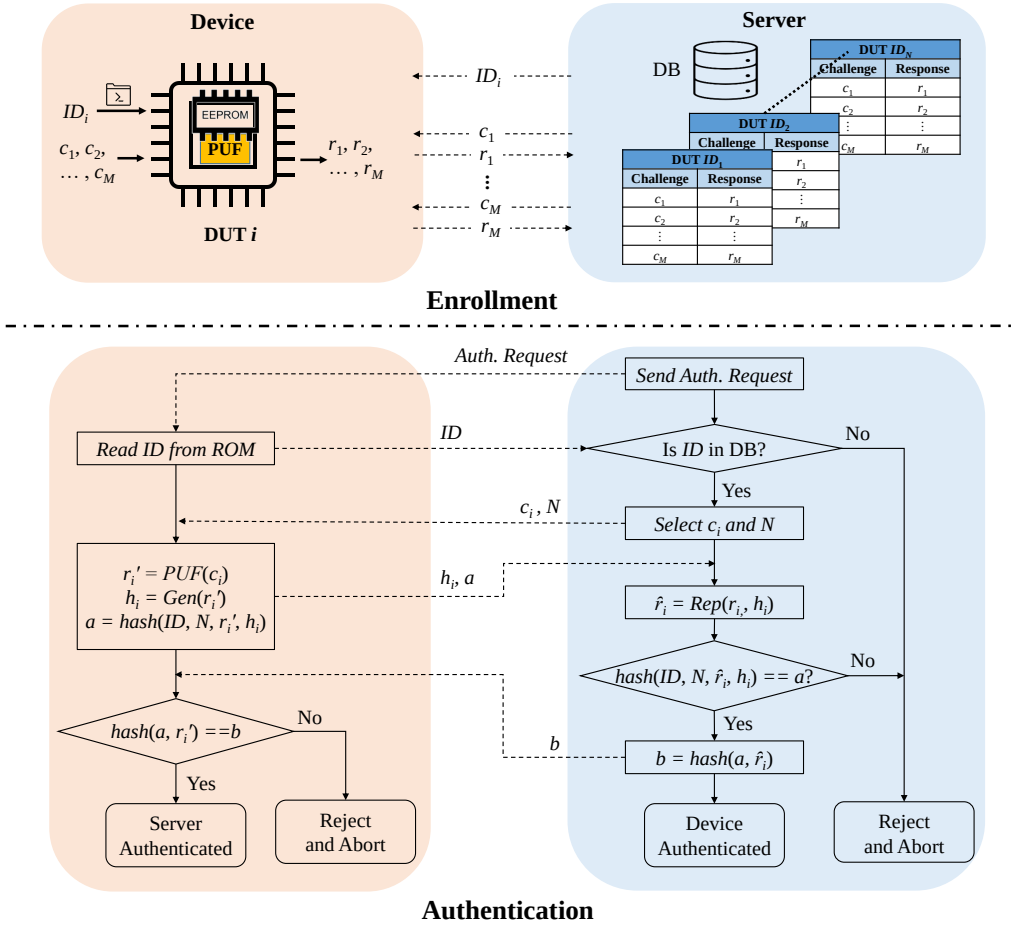

Fig. 5. PUF-based mutual authentication protocol [11]

protocol. The lightweight AES algorithm can also be used for device authentication without resorting to hardware-intensive PKC. However, using symmetric-key cryptography for device authentication requires the secret key to be stored safely in the device's local memory besides its vulnerability to man-in-the-middle (MITM) and replay attacks. Even then, implementing device authentication solutions with AES is still more costly than with PUF. As an illustrative example, a 64-bit arbiter PUF with an LFSR needs about 784 logical gates, which is much less than that required by an AES circuit.

V. Are RFF and PUF Twins?

To address this question, we compare RFF and PUF from two perspectives.

A. Entropy Source

Both RFF and PUF are hidden birthmarks transcribed into a device during its manufacturing process. In contrast to preprogrammed IDs, they are tamper-evident. Attempts to probe into the transceiver or PUF internals will dramatically change their physical characteristics and permanently erase the birthmarks. The uniqueness and unpredictability of RFF and the CRPs of PUF are derived from randomly distributed analog parameters in identically designed circuits, which have larger variance in more advanced semiconductor manufacturing technologies. The uniqueness of a PUF is predominantly determined by the random inter-die process variations. However, correlated intra-die variations may impact the uniqueness of a PUF as they become significant in advanced deep-submicron device technology.

Tighter component tolerances incur a higher cost due to lower manufacturing yield and more expensive tooling. To keep the cost of IoT devices low, designers have to accept the component tolerance provided by the manufacturer for yield control. As a consequence of heterogeneous component parameter deviations in wireless transceivers, RFF requires no extraneous circuits to be added into a transmitter for its identification. Hence, it is applicable to any IoT technologies that have wireless communication capabilities, such as WiFi [7], [8], ZigBee [6], and LoRa [9], to name but a few.

PUF makes use of digital stimuli to extract the parametric mismatches from symmetric cells and digitize them into bits for chip identification. PUF can therefore extract more randomness from microscopic structures not limited to wireless communication components. Unlike RFF, PUF circuits can be implemented on both FPGA and ASIC platforms. In general, the structures are not derived from original functional circuits. Even if a functional module can be used for PUF response generation, its original functionality and performance will be

TABLE I
A COMPARISON OF UNIQUENESS, RELIABILITY AND HARDWARE RESOURCE CONSUMPTION OF DIFFERENT STRONG PUF IMPLEMENTATIONS.

PUF Design	Uniqueness	Reliability	Temp./Volt. Ranges	Technologies	Hardware Consumption
Diode-clamped Inv PUF [12]	49.89%	99.18%	−40°C ∼ 90°C 1.1V±20%	40 nm CMOS	4,719 μm^2
Arbiter PUF (APUF) [10]	23%	95.18% 96.26%	0 ∼ 70°C @1.8V 1.8V±2% @27°C	180 nm CMOS	1,212 $\mu m\times$ 1,212 μm
FFAPUF [13]	41.53%	97.10% 93.90%	0 ∼ 75°C @1.0V 1.0V±10% @27°C	28 nm Artix7 FPGA	128 Slices
Interpose PUF (iPUF) [14]	40%	97.90%	0 ∼ 70°C @1V	28 nm Artix7 FPGA	2× APUF for a (1,1)-iPUF

Ideal uniqueness: 50%, ideal reliability: 100%.

affected. For example, a PUF response can be extracted from a memory array upon reset, but the memory cells cannot be used for PUF operation and data storage simultaneously. The original data will be lost when the array is reset for PUF response generation. As PUFs are natively robust against reverse engineering attacks, security protocols can be devised to protect specific software or hardware blocks using PUF as a root-of-trust for device-specific secret generator in cryptographic operations. Besides device identification and authentication, PUFs found versatile applications in IC camouflaging, logic locking and obfuscation, hardware intellectual property (IP) watermarking and even digital forensic [5].

Entropy analysis provides invaluable insight both into the number of devices that can be classified by RFFI, and into the response uniformity and bit-aliasing of a PUF design. The entropy of RFF is governed by the complexity and performance specifications of the transceivers. As it is difficult or expensive to control the analog imperfections of wireless transceivers, there is very limited room for customizing the transceiver design to boost its entropy for RFF. Conversely, the PUF design is decoupled from the original circuit or system functionality and the response is digitized. To prevent replay attacks in device authentication, a large number of CRPs have to be produced by a strong PUF to ensure that each CRP can be used only once. In the meantime, an adversary may collect previously used CRPs to machine learn a strong PUF and predict its responses to unseen challenges. Hence, the entropy harvested by a specific implementation of a PUF is critically important to its uniqueness and security. Fortunately, the uniqueness and randomness of native PUF responses can be enhanced by the customized layout of symmetric cell structures and randomness extractors at the cost of small hardware overhead. Entropy analysis is particularly beneficial for boosting the model-building resistance of PUF against machine learning attacks, which will be discussed in Section VII.B.

B. Protocol Implementation

The identification data of RFF and PUF, in the form of a learning model or binary templates, have to be collected in a secure and controlled environment before they can be used for device authentication. The templates or device models are stored in a secure server database.

RFF is inherent in the transmitted signals. There is neither hardware nor processing overhead at the device (prover) side. By identifying the transmitter through RFF, the provenance of the received data is automatically validated. This is beneficial as the audit trait of the received packet thwarts MITM and replay attacks. However, when an IoT device authenticates the server, there will be overheads imposed on the RFF extractor and challenges in safekeeping the RFF template or the model learned. Therefore, RFF is mostly used for unilateral authentication. In addition, RFF extraction can also be affected by the quality of low-cost receivers [2]. Finally, because commercial off-the-shelf wireless transceivers do not make raw IQ samples available to developers, RFFI requires specialized receivers, such as the USRP SDR platforms [9].

The entire CRP set of a PUF is intrinsically embodied in its nano-structure upon fabrication. There is no safe-keeping issue, since the direct CRP measurement interface can be permanently disabled after enrollment. Both the DUT and server can act as prover as well as verifier. Hence, mutual authentication is feasible for PUF. After several message exchanges, either entity can prove that it knows the response to a random challenge picked by the other without divulging any CRPs, as illustrated in Fig. 5. These message exchanges used for authentication can be performed in any digital channel without being impacted by the quality of the channel or transceivers. The authentication messages have high noise immunity, since they are digitally encoded and the errors can be readily reconciled by error correction codes (ECCs).

It is envisaged that the complementary functions of RFF and PUF will alleviate the limitations of classical key-based cryptography in strengthening wireless network security against emerging threats.

C. Epilogue

Both RFF and PUF stem from the random semiconductor manufacturing process variations. They are manifested as unique device identities in the RF and digital domains, respectively. As both the input challenge and output response of a PUF are digital words, they can be readily extracted and manipulated by logic circuits for CRP obfuscation, reliability enhancement and entropy boosting. Both identities generated by RFF and PUF need to be extracted in a pre-enrolled process before they can be used by the authentication protocol

to identify the associated devices in the field. From these perspectives, they can be considered twins.

VI. Are RFF and PUF Competitors or Allies?

To have the best of both worlds, is it feasible to forge an RFF-PUF alliance? Indeed, very much so! Hence we propose a cooperative RFF-PUF mutual authentication and key establishment protocol. Specifically, RFF is complemented by PUF for mutual authentication and PUF is augmented by RFF for resistance against MITM and modeling attacks.

To elaborate a little further on our proposal, the DUT is equipped with a PUF and a wireless transceiver. The CRPs of the PUF and RFF of each registered device are enrolled in the server's database *DB* using its MAC address as the device ID. The ID does not have to be encrypted to protect the transmitter and device fingerprints against traceability attack. This is because the ID is not used in the generation of PUF's CRPs, neither is it relied upon by the RFF. Indeed, the RFF and CRPs are kept private in the well protected server's databases. Collection of openly transmitted IDs of different transmitters provides no knowledge about the credentials (RFF and CRP) of any wireless devices. For each connection request, the DUT will transmit an initiation packet that contains its MAC address in the header, plus a nonce N, a randomly selected challenge c_i and a helper data h_i in the payload, where the helper data $h_i = \mathit{Gen}(r_i')$ is generated by the ECC encoder from the measured PUF response $r_i' = \mathit{PUF}(c_i)$ to the challenge c_i. Due to environmental variations, the response r_i' generated by the PUF may not be identical to the enrolled response r_i stored in *DB*. Note that the PUF response is not transmitted. Only the challenge and the helper data are transmitted. To forge the source address or to tamper with the message, the attacker must intercept and re-transmit the message using its own transmitter. The server can extract the RFF from the received packet to validate if the packet is sent from the transmitter of the registered DUT with the claimed MAC address. If the RFF does not match the enrolled RFF for the MAC address, the server will reject the DUT. Otherwise, the server retrieves the enrolled PUF response r_i to c_i from *DB* and sends the DUT a proof $b = hash(\mathit{MAC}, N, \hat{r}_i, h_i)$ for confirming that it knows the shared secret response without divulging it, where $\hat{r}_i = \mathit{Rep}(r_i, h_i)$ is the response reproduced by the ECC decoder using the received helper data h_i and r_i. To authenticate the server, the DUT computes $a = hash(\mathit{MAC}, N, r_i', h_i)$. It accepts the server if a matches the received b and rejects the server otherwise. This scheme requires only one message exchange for mutual authentication. The nonce N ensures the freshness of the proof b for each authentication session for preventing replay and MITM attacks.

More importantly, both the server and the DUT establish a shared secret, since $r_i' = \hat{r}_i$ if and only if they are both genuine. This shared secret can be used as a session key for subsequent encrypted communications between them without requiring further message exchange and any computationally intensive PKC algorithms. In summary, the proposed RFF-PUF allied protocol achieves secure mutual authentication with key exchange (MAKE). It requires even less message transfer than the family of PUF-based mutual authentication protocols without key exchange. Compared to known MAKE protocols, it does not rely on the storage of historic session secrets and pseudonyms for session key updates. It does not respond to authentication requests from unauthorized transmitters without a matched RFF, even if the message payload carries a valid ID. Hence it prevents MITM, replay, desynchronization and modeling attacks at the root.

VII. Prospective Challenges

A. Reliability

Existing RFFI schemes are seldom evaluated against supply voltage and temperature variations. However, the carrier frequency of low-specification RF oscillators may drift with temperature [9]. As RFF extraction is carried out in the analog domain, this drift can be mitigated by compensation and calibration. By contrast, the wireless channel impulse response is a unique environmental factor impacting RFF, since $h(t)$ in (1) fluctuates during device deployment. The variations may be significant enough to overshadow the influence of the RFF determined during enrollment [15]. This problem may be addressed with the aid of deep learning using data augmentation emulated under different channel conditions during enrollment. Employing channel-agnostic features for training and isolating the channel characteristics from the RFF by signal processing algorithms are also worth investigating. Finally, existing RFF schemes have been conceived for single-hop scenarios. The challenge in multiple-hop scenarios lies in the difficulty in reliably extracting transmitter characteristic deviations from the wireless channel between the transmitter/receiver and the relays.

The reliability of PUF has been widely studied with significant progress made over the last decade. Increasing (decreasing) the supply voltage increases (decreases) the switching speed of a transistor. Similarly, increasing (decreasing) the temperature reduces (raises) the threshold voltage and carrier mobility of a transistor. Lowering the threshold voltage speeds up the switching of a transistor, while reducing the carrier mobility slows it down. Depending on its biasing, one of these effects dominates and a transistor may exhibit negative or positive temperature coefficient. Many silicon-based PUFs are designed based on the timing race of symmetric transistor networks. Any voltage and temperature fluctuations may result in some bit flips for the response generated by the PUF in the field. Depending on the CRP space of a PUF and on the number of devices to be identified, a few response bit errors may be acceptable and a Hamming distance threshold can be set in authentication to reduce false rejection with a very low risk of false acceptance. Other low-cost reliability enhancement techniques include unreliable response filtering, temporal and spatial majority votings, etc. To significantly reduce the bit error rate of the raw PUF responses over a broad spectrum of operating conditions, ECC is required, as illustrated in Fig. 5. A less reliable PUF will require a stronger ECC and more bandwidth to transmit the helper data.

B. Security

As RFF is still in its nascent stage, ongoing research focuses mainly on improving its classification accuracy and on exploring RFF for securing wireless networks against Sybil attacks, rather than attacking RFF for impersonation. Similarity-based RFFI schemes are more prone to spoofing attacks, because an attacker can estimate the RFF features and then use digital predistortion to emulate an authentic device. By contrast, deep learning-based schemes do not extract specific features but use raw IQ samples, making spoofing attack success unlikely. However, it is generally believed that guessing the RFF of a transmitter for any packet transmission is practically infeasible, when combined features of multiple RF components are used to train the classifier and the features selected are not known by the attackers.

Due to the minuscule imperfections of the semiconductor production process, even if a PUF is known down to the atomic level, it remains prohibitively difficult to re-fabricate it perfectly to reproduce the same CRPs. This physical unclonability does not, however, prevent the black-box one-way mapping from being modeled. The large CRP space of strong PUFs allows adversaries to collect enough used CRPs for learning, so that the trained network can predict the responses to unseen challenges with high accuracy. Strong PUF architectures constructed for example by the arbiter PUF of Fig. 4 are vulnerable to modeling attacks. A potential solution is to avoid leaking the CRP during authentication, and to prevent the responses from being measured directly except during enrollment. In Fig. 5, a syndrome construction based implementation of $Gen(r)$ is used to ensure that r is hard to predict, even if multiple helper data can be obtained from the noisy variants of r [11]. However, recent research has shown that public helper data can be exploited to model a PUF. One way to circumvent such attacks is to improve the reliability of the PUF so that less helper data is available to the attacker. Another way is to increase the modeling complexity of the PUF or use a more complex code structure to increase the amount of helper data required for a successful attack. More effective recent approaches rely on reconfiguring the CRPs, or poisoning the training data collected by adversaries for rendering their clone PUF models unusable.

VIII. Conclusion

We advocate two emerging lightweight techniques for verifying the connected device's identity. Explicitly, RFF and PUF are inherent device fingerprints in the RF and digital domains, respectively, which strike a compelling threat vs. complexity trade-off, but also have their limitations. On this basis, we conclude that RFF and PUF are twin techniques. We also demonstrated that they can form a powerful alliance by combining their respective best features in a cooperative mutual authentication and key establishment protocol.

Acknowledgement

The work of J. Zhang was supported by Royal Society Research Grants under grant ID RGS/R1/191241. The work of C. H. Chang was supported by the Singapore Ministry of Education AcRF Tier 2 grant No. MOE-T2EP50220-0003. The work of L. Hanzo was supported by the EPSRC projects EP/N004558/1, EP/P034284/1, the Royal Society's GCRF Grant as well as the European Research Council's Advanced Fellow Grant QuantCom.

References

[1] N. Wang, W. Li, P. Wang, A. Alipour-Fanid, L. Jiao, and K. Zeng, "Physical layer authentication for 5G communications: Opportunities and road ahead," *IEEE Network*, vol. 34, no. 6, pp. 198–204, 2020.

[2] J. Zhang, R. Woods, M. Sandell, M. Valkama, A. Marshall, and J. Cavallaro, "Radio frequency fingerprint identification for narrowband systems, modelling and classification," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 3974 – 3987, 2021.

[3] C. H. Chang, Y. Zheng, and L. Zhang, "A retrospective and a look forward: Fifteen years of physical unclonable function advancement," *IEEE Circuits Syst. Mag.*, vol. 17, no. 3, pp. 32–62, Aug. 2017.

[4] Q. Wang and G. Qu, "A silicon PUF based entropy pump," *IEEE Trans. Depend. Sec. Comput.*, vol. 16, no. 3, pp. 402–414, May 2019.

[5] Y. Zheng, Y. Cao, and C. H. Chang, "A PUF-based data-device hash for tampered image detection and source camera identification," *IEEE Trans. Inf. Forensics Security*, vol. 15, no. 7, pp. 620–634, Jul. 2019.

[6] L. Peng, A. Hu, J. Zhang, Y. Jiang, J. Yu, and Y. Yan, "Design of a hybrid RF fingerprint extraction and device classification scheme," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 349–360, 2018.

[7] V. Brik, S. Banerjee, M. Gruteser, and S. Oh, "Wireless device identification with radiometric signatures," in *Proc. 14th ACM Int. Conf. Mobile Computing Networking*, San Francisco California USA, Sep. 2008, pp. 116–127.

[8] K. Sankhe, M. Belgiovine, F. Zhou, L. Angioloni, F. Restuccia, S. D'Oro, T. Melodia, S. Ioannidis, and K. Chowdhury, "No radio left behind: Radio fingerprinting through deep learning of physical-layer hardware impairments," *IEEE Trans. on Cogn. Commun. Netw.*, vol. 6, no. 1, pp. 165–178, 2019.

[9] G. Shen, J. Zhang, A. Marshall, L. Peng, and X. Wang, "Radio frequency fingerprint identification for LoRa using deep learning," *IEEE J. Sel. Areas Commun.*, 2021.

[10] D. Lim, J. W. Lee, B. Gassend, G. E. Suh, M. Van Dijk, and S. Devadas, "Extracting secret keys from integrated circuits," *IEEE Trans. VLSI Syst.*, vol. 13, no. 10, pp. 1200–1205, 2005.

[11] A. van Herrewege, S. Katzenbeisser, R. Maes, A.-R. Sadeghi, I. Verbauwhede, and C. Wachsmann, "Reverse fuzzy extractors: Enabling lightweight mutual authentication for PUF-enabled RFIDs," in *Proc. Int. Conf. Financial Cryptography and Data Security*, Kralendijk, Bonaire, Feb. 2012, pp. 374–389.

[12] Y. Cao, C. Q. Liu, and C. H. Chang, "A low power diode-clamped inverter-based strong physical unclonable function for robust and lightweight authentication," *IEEE Trans. Circuits Syst.*, vol. 65, no. 11, pp. 3864–3873, 2018.

[13] C. Gu, W. Liu, Y. Cui, N. Hanley, M. O'Neill, and F. Lombardi, "A flip-flop based arbiter physical unclonable function (APUF) design with high entropy and uniqueness for FPGA implementation," *IEEE Trans. Emerg. Topics Comput.*, vol. 9, no. 4, pp. 1853 – 1866, Oct.-Dec 2021.

[14] P. H. Nguyen, D. P. Sahoo, C. Jin, K. Mahmood, U. Rührmair, and M. van Dijk, "The interpose PUF: Secure PUF design against state-of-the-art machine learning attacks," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pp. 243–290, 2019.

[15] S. D'Oro, F. Restuccia, and T. Melodia, "Can you fix my neural network? real-time adaptive waveform synthesis for resilient wireless signal classification," in *Proc. IEEE INFOCOM*, 2021, pp. 1–10.